Intent-driven, fully automated deployment of anycasted load balancers with HAProxy and Python

DENOG 11

Maximilian Wilhelm

Agenda

1. Who's who
2. Context
3. The past
4. The Idea
5. The now
6. Q & A

Who's who

Maximilian Wilhelm

- Networker
- OpenSource Hacker
- Fanboy of
  - (Debian) Linux
  - ifupdown2
- Occupation:
  - By day: Senior Infrastructure Architect, Uni Paderborn
  - By night: Infrastructure Archmage, Freifunk Hochstift
  - In between: Freelance Solution Architect for hire
- Contact
  - @BarbarossaTM
  - max@sdn.clinic

Context

Context

Context

- Paderborn University
  - 20.000 students
  - 2.500 employees
- Lots of central IT services
  - IDM (LDAP, Kerberos, AD, …)
  - Mail (SMTP, IMAP, PMX, Mailman, Exchange)
  - An aweful lot of websites
  - eLearning things (Moodle, PAUL, …)
  - SharePoint
  - File services
  - The Internet
  - …

Who's who

Context

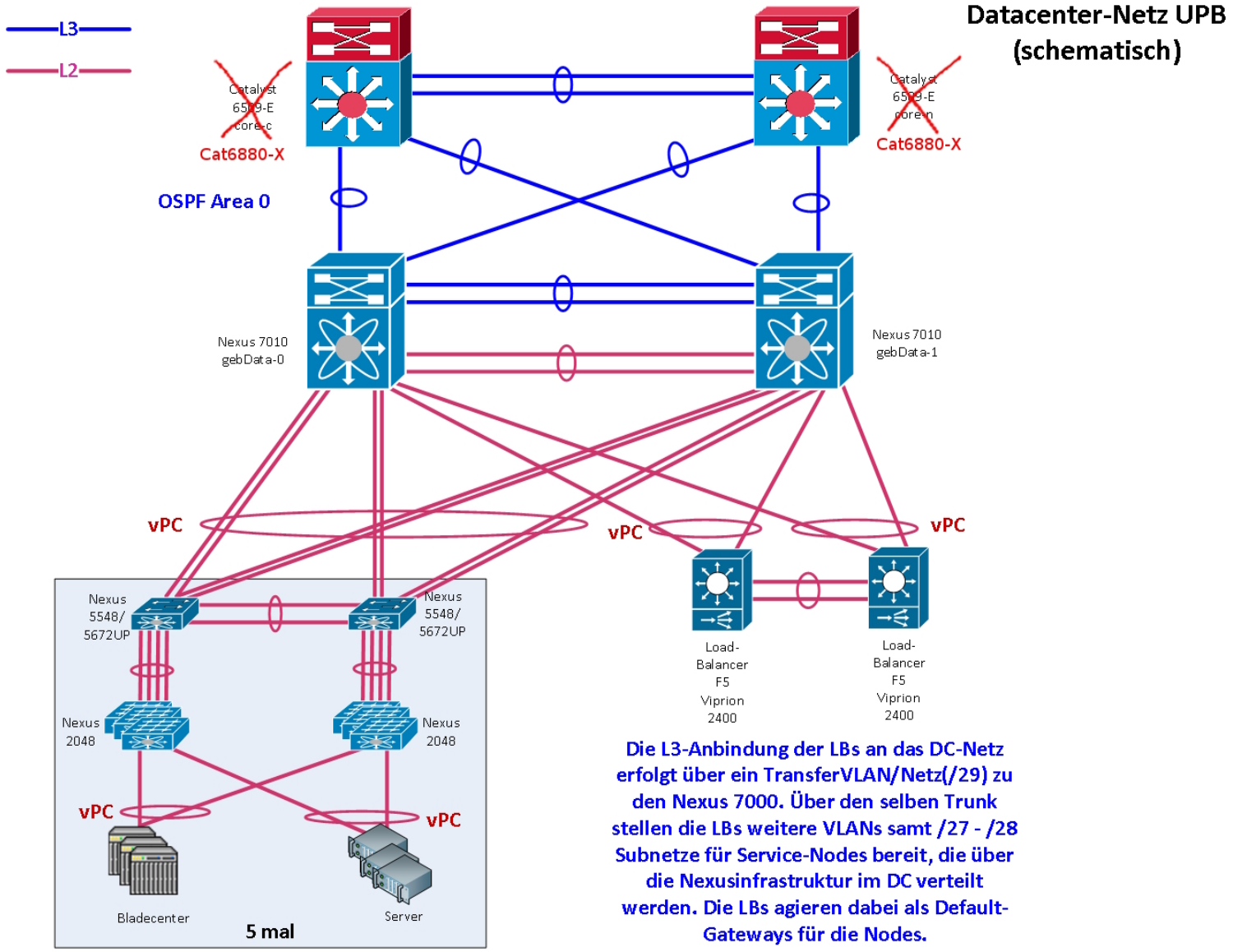
The Past

The Past

Who's who

Context

The Past



The Past

- Cisco Nexus based L2 fabric
  - VLANs for service / backend networks
- 2x F5 Viprion 2400 LBs
  - Router / default gateway for all service networks
  - Prefixes for VIPs statically routed to VRRP IP
  - Prefixes for backend networks statically routed to VRRP IP
  - No ACLs between service networks
  - Out-of-everything end of 2018
- Manually configured
  - Even monitoring

Who's who

Context

The Past

Who's who

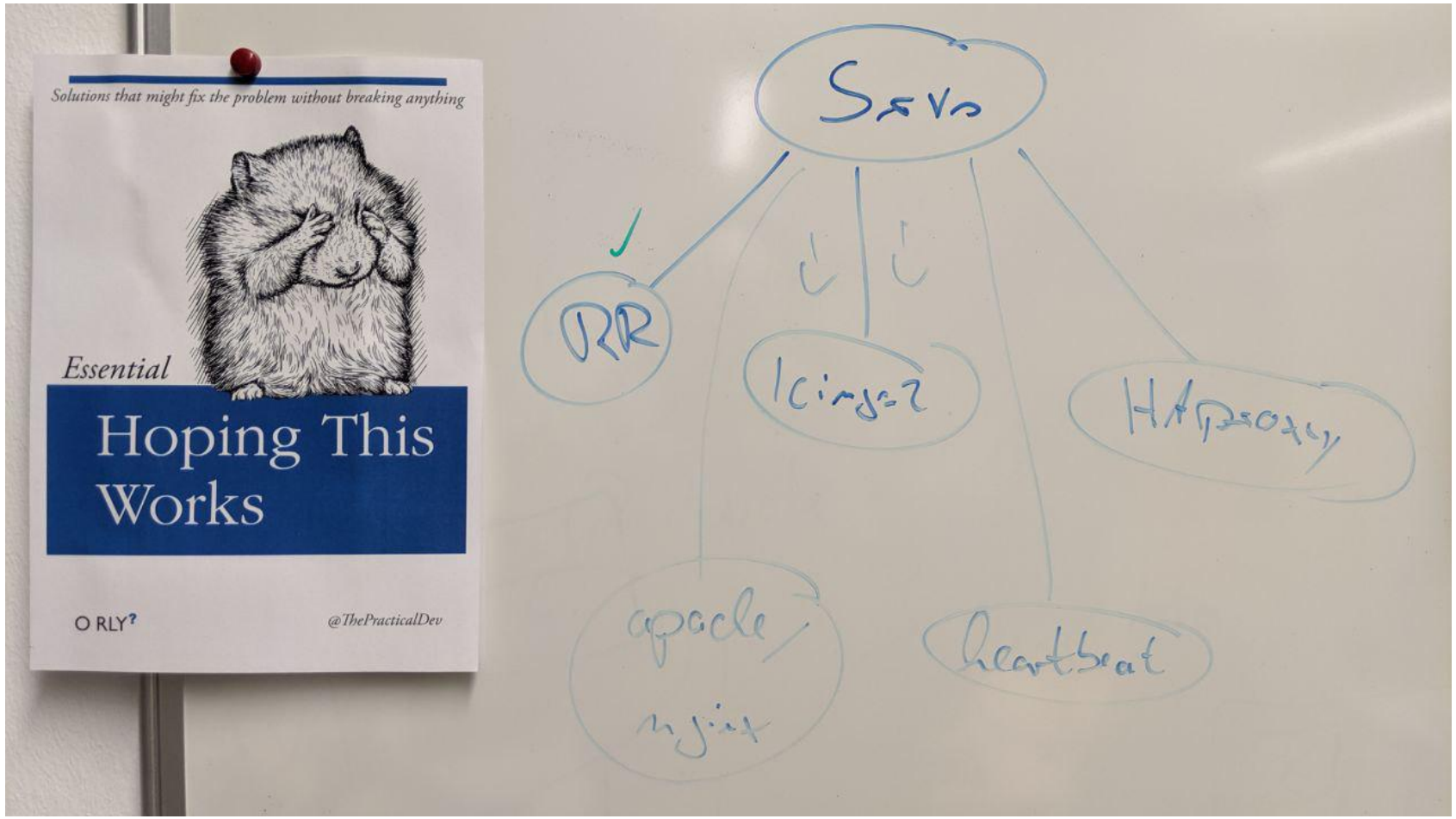
Context

The Past

The Idea

The Idea

The big picture



The idea

- A ***service*** as central config element
- Can be balanced by
  - Anycast
  - HAProxy
- If balanced
  - Service VIPs announced via BGP
  - Should be Active/Active
- Monitoring configured automatically
  - Checks for frontends / VIPs as well as backends
- Config of webserver(s) generated
- Should additionally allow
  - H/A clusters
  - Caching layer for web stuff
- Subnets of service nodes should be routed by DC routers
  - with ACLs

What was in the cards?

Working DC network setup

- All VLANs everywhere
- BGP capable DC routers

Heavy automation for Linux boxes

- bcfg2
- Written in Python
- Easily extendable
- Config generators for Icinga2
- Basic Apache2 templating

People not afraid of automation

- On the contrary

Now what IS a *service*

- Has an FQDN
  - resolves to IP and/or Legacy-IP addresses
- Has a proto and service
  - proto derived from service, if possible
  - e.g. *tcp/http* or *tcp/80*
- Is provided by hosts of *$bcfg2_group*
  - e.g. *kdc-production*
- May be *anycasted*
- May be *balanced*
  - And the LBs anycasted
- May be a web thing
  - With special http config
  - e.g. template, redirects and stuff
- May have special monitoring config

How does it look like?

```
mwilhelm@kili:/bcfg2/etc/services/imt/infrastructure/anycasted$ cat kerberos-kdc.srv

anycast: True
status: produktiv

name: kerberos-kdc

fqdn: kerberos.srv.imt.uni-paderborn.de
service: kerberos

bcfg2_srv_group: kdc-slave

monitoring:
  virtual_bcfg2_groups:
    - kdc
    - imt-master
```

Well OK, it has a defaulting mechanism, too

```
mwilhelm@kili:/bcfg2/etc/services/imt/infrastructure/anycasted$ cat defaults.yaml

anycast: True
status: produktiv
```

```
mwilhelm@kili:/bcfg2/etc/services/imt/infrastructure/anycasted$ cat kerberos-kdc.srv

name: kerberos-kdc

fqdn: kerberos.srv.imt.uni-paderborn.de
service: kerberos

bcfg2_srv_group: kdc-slave

monitoring:
  virtual_bcfg2_groups:
    - kdc
    - imt-master
```

Who's who

Context

The Past

The Idea

The Now

The Now

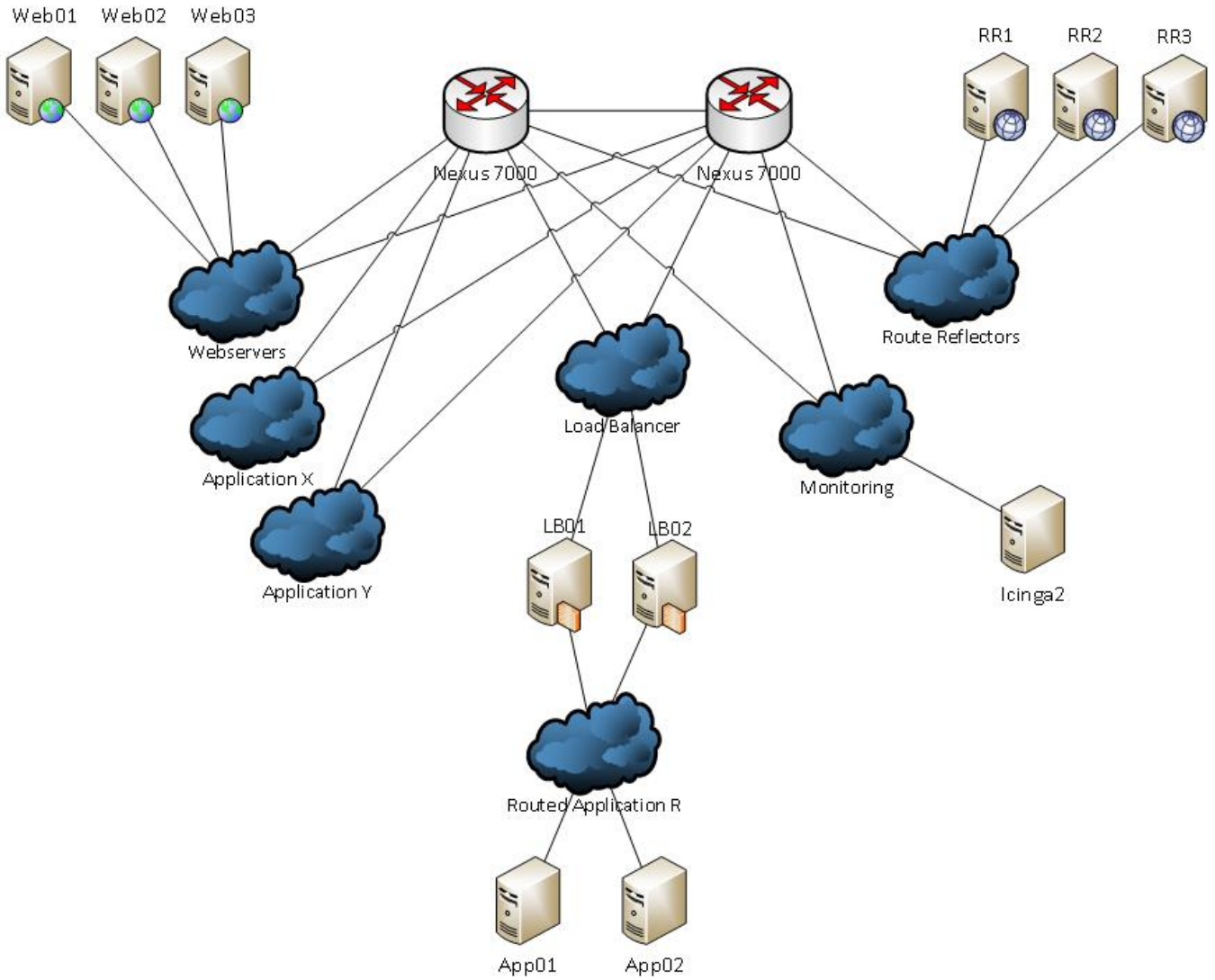
Who's who

Context

The Past

The Idea

The Now



Lessons learned

Bad NIC firmware is bad

- BGP timeouts are long
- Recovery times are bad when L2 is a black hole
- BFD will solve this

HAProxy configuration is complex

- Lots of switches have effect on other switches
- No way to ask HAProxy what config options are active

The good

Backends with support for Proxy Protocol

- Apache2
- Cyrus IMAP
- Dovecot
- Exim
- Nginx
- Postfix
- Varnish
- …

The bad

OpenLDAP

- No support for Proxy Protocol
- Has to be DNATed by HAProxy when slapd should see client IPs
- Therefore LDAP backends have to be routed by HAProxy

Exchange

- Funny problems with timeouts (solved)
- Funny problems with Outlook for Mac clients

SharePoint

- Funny problems when you don't use tcp mode for some vHosts
- I want this hour of my life back

Bonus level: Packet filter configuration

- We know what ports a service is using
- We know where (backend, frontend)
- Let's generate netfilter rules
- Limiting access to source prefixes just came on top
- Specifying *additional_ports*, too

```
mwilhelm@kili:/bcfg2/etc/services/imt/infrastructure/anycasted$ cat proxy.srv

name: proxy

fqdn: proxy.srv.imt.uni-paderborn.de
service: proxy
protos: tcp
port: 3128

bcfg2_srv_group: proxy-server-produktiv

acl:
  allow_from:
    - imt_thinclients
    - imt_fw_mgmt
```

Further Reading

BGP / networking basics

- https://myfirst.network

Anycast with Cisco Nexus 7000 and Debian Linux

- https://blog.sdn.clinic/2018/02/anycasted-services-with-debian-bird-anycast-healthchecker-and-cisco-nexus-7000/

Anycast all the things

- https://www.slideshare.net/BarbarossaTM/anycast-all-the-things

Questions?